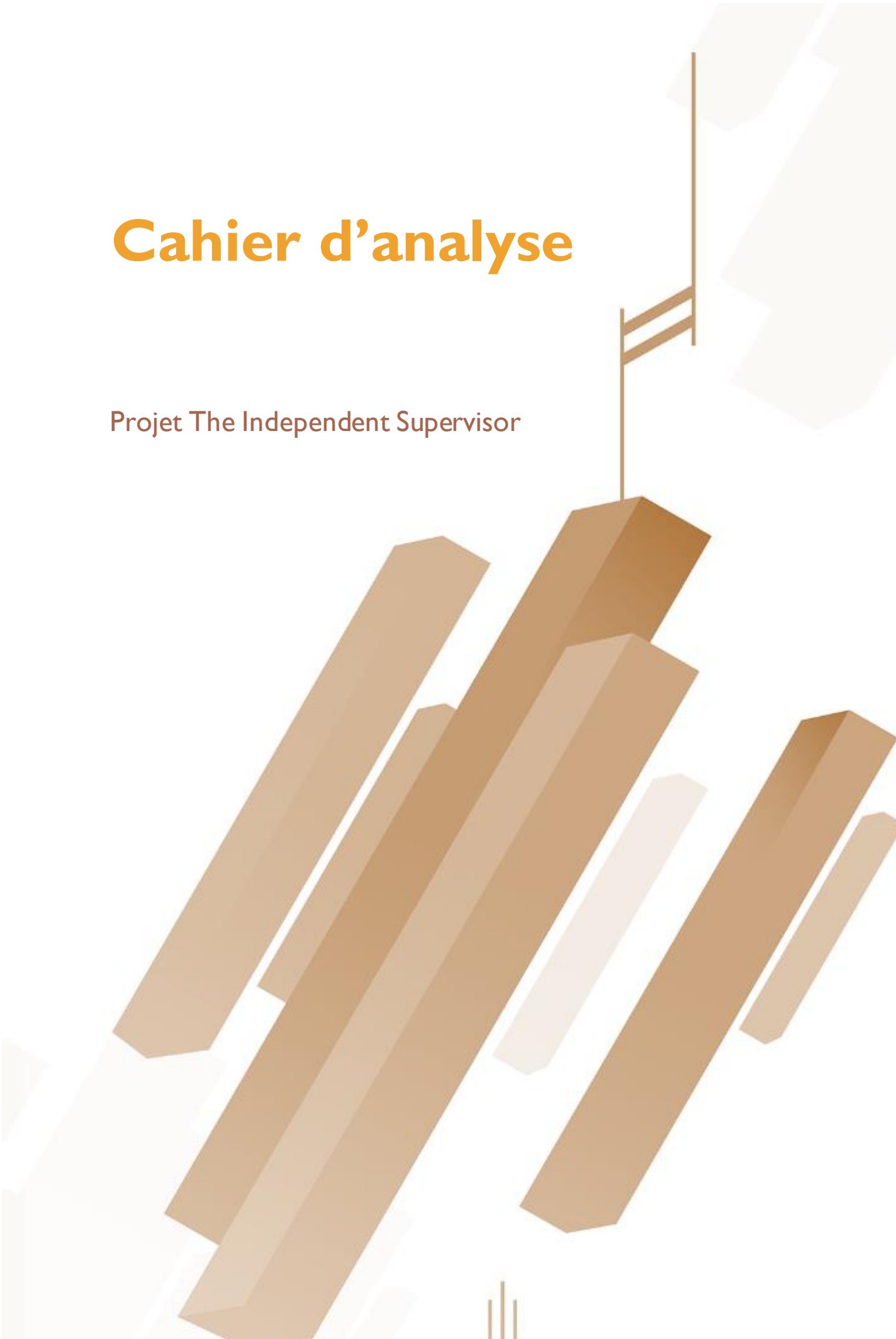


# Cahier d'analyse

Projet The Independent Supervisor



## Table des matières

I.	Introduction .....	4
1.	Contexte du projet .....	4
2.	Objectif général du projet.....	4
1.	Architecture .....	5
a)	Existant.....	5
b)	Objectif .....	6
II.	Analyse et Modélisation .....	7
1.	Modélisation des interfaces utilisateur & UML.....	7
a)	Utilisateur .....	8
b)	Administrateur .....	18
c)	Interactions avec le parc d'application.....	24
d)	Vue mobile .....	30
e)	Diagramme de classes .....	31
2.	MEAN Stack.....	32
a)	MongoDB.....	33
b)	Angular .....	35
c)	Node.JS .....	38
d)	Express .....	39
e)	Pourquoi ce choix ? .....	40
3.	Serveur dédié .....	41
a)	Hébergement.....	41
b)	Architecture .....	42
c)	Fiabilité et Vulnérabilité .....	43
d)	Déploiement .....	43
e)	Maintenance .....	43
f)	Communication.....	43

Commentaire	Date de modification
Création du document	13/11/2018
Prévision de la partie Analyse & Modélisation	26/11/2018
Ajout de plusieurs diagrammes de séquence	27/11/2018
Ajout des derniers diagrammes de séquence	03/12/2018
Descriptions des technologies qui correspondent aux besoins	08/01/2019
Termine les descriptions des technologies	09/01/2019
Description de l'hébergement	09/01/2019
Ajout Fiabilité	09/01/2019
Ajout des mockups	09/01/2019
Mise au point MOE & MOA	11/01/2019
Ajout des modifications suite à la mise au point	13/01/2019

// WIP

1. Check orthographe
2. Remplacement du mot « Sauvegarde données » par Sauvegarde de données (diagramme modification de configuration)

# I. Introduction

## 1. Contexte du projet

Dans le cadre de notre formation en école d'ingénieurs, nous devons réaliser un projet de fin d'études durant notre 5<sup>e</sup> année. Ce projet débute mi-Septembre 2018 correspondant à notre début de période école pour cette dernière année, pour finir à la fin de notre période école, pendant le mois de Février 2019. Étant donné que le projet se déroule en même temps que la période école, il n'est pas possible de travailler à plein temps sur le projet, une demi-journée sera donc consacrée au minimum chaque semaine.

The Independent Gamers est une équipe de joueurs amateurs, cofondé en 2013 par Brandon SIMON-VERMOT, cette équipe partage l'idée de fournir une ambiance divertissante en proposant diverses activités et la possibilité de rencontrer des joueurs réguliers sur une panoplie de jeux divers. The Independent Supervisor est le fruit d'une idée venant du chef de l'équipe de joueurs amateur qui possède un serveur dédié, sur lequel plusieurs serveurs de jeux sont installés.

- Brandon SIMON-VERMOT : Maitre d'œuvre
- Aneur SOUKHAL : Encadrant de projet & maitre d'ouvrage

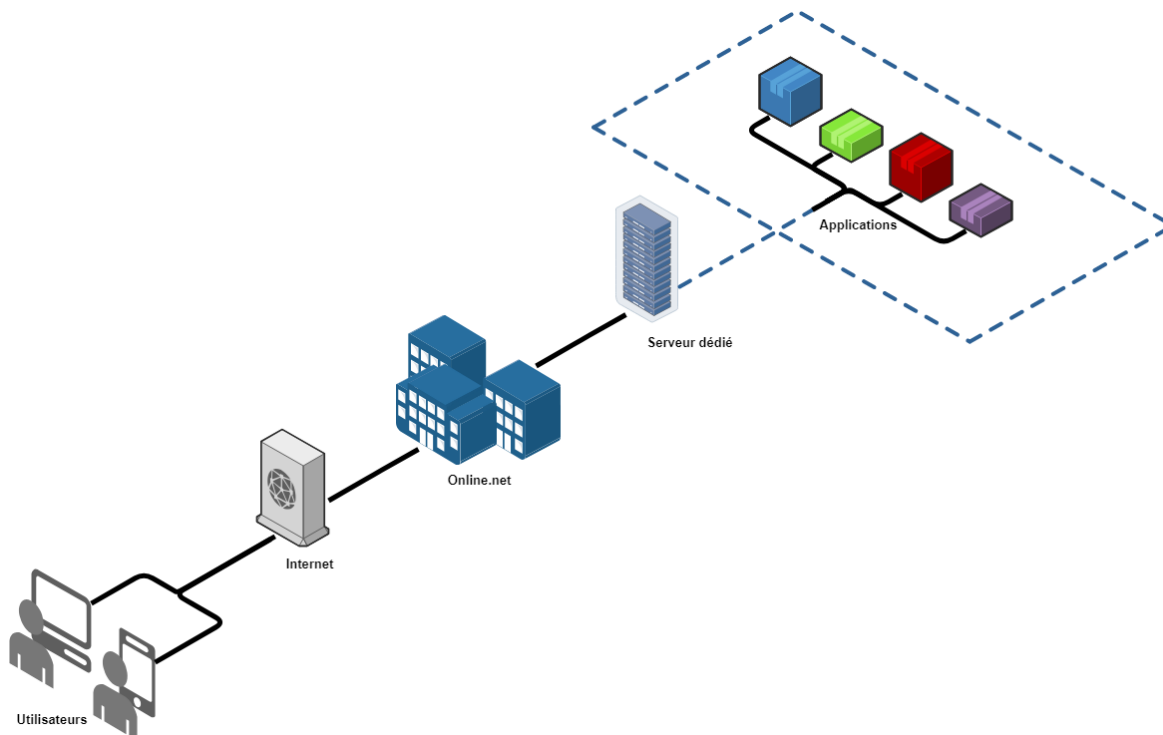
Pour plus d'informations de manière globale, la lecture du cahier de spécification peut s'avérer utile à la bonne compréhension du projet.

## 2. Objectif général du projet

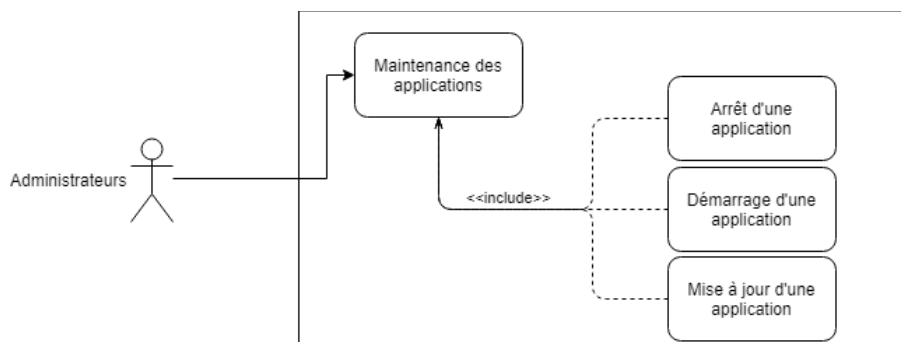
Le besoin de ce projet est de superviser les serveurs de jeux et offrir la possibilité de les mettre à jour automatiquement, de les maintenir. Pour ce faire, des lancements, des arrêts d'application (via des commandes enregistrées associées à une application par exemple) seront exécutés automatiquement pour remplacer la maintenance manuelle, ce qui est fait actuellement. Ce projet part de zéro, aucun site Web et les outils liés (serveur Web, BDD, ...) ne sont déjà installés sur le serveur dédié, seuls les serveurs de jeux sont actuellement présents. Cette application web permettra donc de pouvoir exécuter une application ou l'arrêter grâce aux informations enregistrées par les administrateurs de l'application Web. Notamment les informations autour de chaque serveur de jeu (emplacement du dossier du serveur, commande de lancement, commande de sauvegarde), qui ici dans le cadre de ce projet seront les applications du parc soumis à la supervision.

## 1. Architecture

### a) Existant

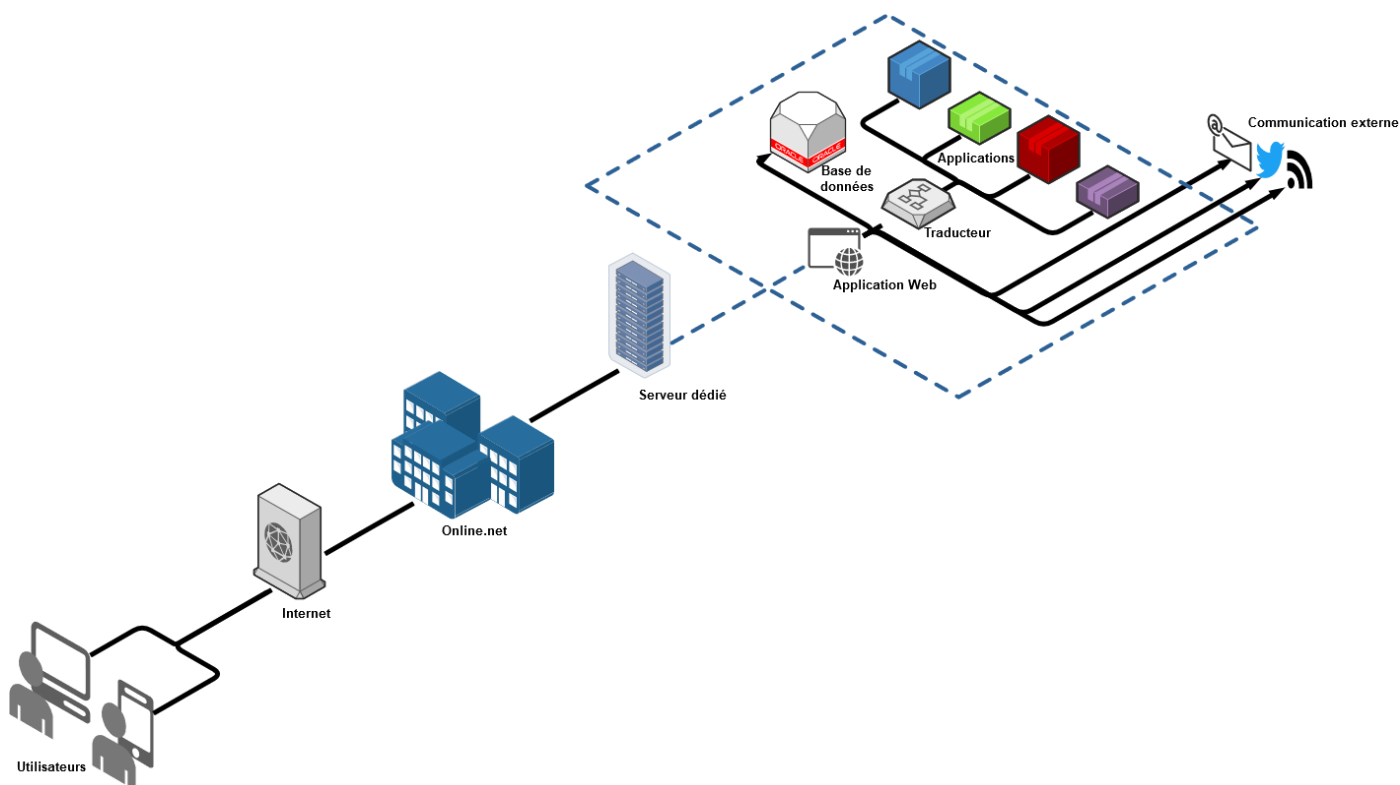


Chaque serveur de jeu, chaque application, présent sur le serveur dédié, qui est actuellement sous Ubuntu, a donc besoin d'être maintenu, d'être redémarré s'il y a une erreur, via des lignes de commandes. Cela demande du temps et un accès au serveur via une application de télémaintenance ou via un hyperviseur, actuellement KVM, ce sont les actuelles limites de l'existant.



Le cas d'utilisation actuel, ne prends pas en compte les joueurs, puisque leurs seules interactions se limitent à la connexion et la communication avec les applications qui sont démarrées sur le serveur dédié. Dans le cadre actuel, il n'y a qu'un seul administrateur qui puisse se connecter sur le serveur dédié pour réaliser la maintenance du parc d'application.

## b) Objectif



Pour mieux représenter le projet, ci-dessus se trouve un schéma comprenant les différents acteurs et éléments constituant l'architecture générale du système envisageable à la fin du projet. Dans un premier temps nos acteurs seront nos utilisateurs (administrateurs ou non), ils devront avoir une connexion Internet pour accéder à l'application Web, cette application est hébergée sur un serveur dédié appartenant au fournisseur Online.net.

L'application Web étant le cœur du projet, elle est associée à différents éléments, une base de données contenant les différentes configurations de nos applications, dans le cadre de ce projet ce seront des serveurs de jeux. Ces applications sont différentes ce qui indique qu'elles peuvent avoir un moyen de communication différent, d'où l'intérêt d'avoir un élément qui nous servira de traducteur, dans le but de transmettre correctement l'information souhaitée.

Cette même application Web permettra de communiquer des informations sur l'état actuel des serveurs de jeux, pour cela il devra être possible de communiquer via différents moyens afin d'en informer nos utilisateurs, via par exemple le réseau social Twitter ou encore un flux RSS.

L'application Web devra pouvoir envoyer des mails aux administrateurs pour les prévenir en cas de problème mais aussi pour toute demandes et suggestions envoyés par nos utilisateurs via l'application Web.

En ayant la possibilité d'accéder au serveur dédié via une application web, depuis un ordinateur ou un smartphone, plusieurs contraintes sont éliminées. En associant ceci avec la possibilité de pouvoir gérer le parc d'applications sans lignes de commandes, cela fait gagner un gain de temps crucial.

## II. Analyse et Modélisation

### 1. Modélisation des interfaces utilisateur & UML

Dans le cas d'un visiteur non identifié, il n'aura pas accès au reste du site web.

Menu	The Independent Supervisor
	<div><div>S'inscrireSe Connecter</div><div>Connexion</div><div>Nom d'utilisateur<input type="text"/></div><div>Mot de passe<input type="password"/></div><div>ConnexionCréer un compte</div></div>

Le visiteur pourra seulement se connecter ou réaliser une demande d'inscription sur le site avec les boutons suivants "Se Connecter" et "S'inscrire".

Ici nous avons aussi l'interface permettant à un visiteur de se connecter avec son nom d'utilisateur et son mot de passe. Si le visiteur ne possède pas de compte, il peut cliquer sur le bouton "Créer un compte", ce qu'il l'enverra vers une interface sensiblement différente de celle permettant de modifier ses informations.

### a) Utilisateur

Pour tous les diagrammes présents ci-dessous, une authentification est nécessaire au préalable, toutes personnes authentifiées auront accès à ces interfaces utilisateurs et pourront réaliser les actions en conséquence sauf indication contraire.

#### Liste des applications

Menu

- Page 1
- Page 2
- Page 3
- ...

The Independent Supervisor

Bienvenue, Mr. X

Mon Compte

Statut	Image	Nom	Description	Action
		...	...	
		...	...	
		...	...	
		...	...	
		...	...	
		...	...	
		...	...	
		...	...	
		...	...	
		...	...	

Page 1 ... N

Ce mockup représente qu'une fois notre utilisateur connecté, il pourra accéder à la page d'accueil du site avec l'exposition de toutes les applications sous forme de liste avec les informations suivantes :

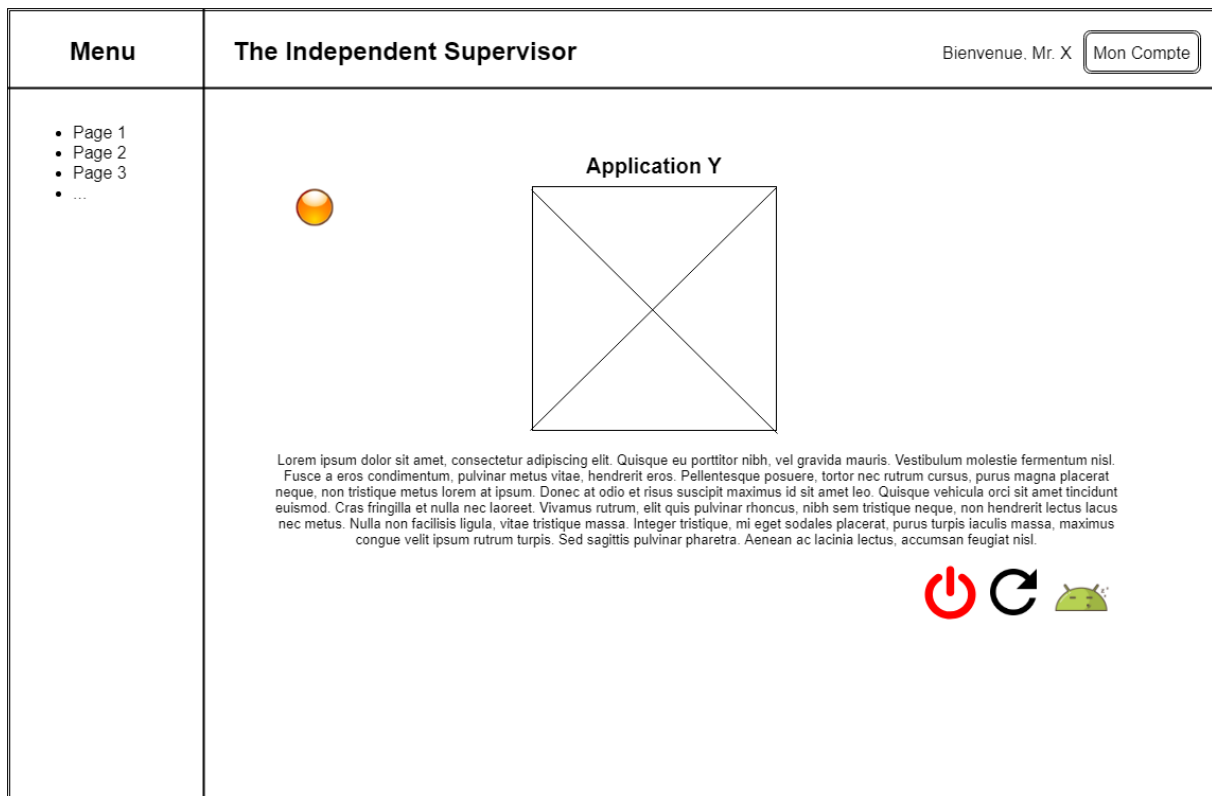
- Le statut de l'application (arrêt, marche, redémarrage, mise sous supervision), représenté ici par une LED de couleur en fonction du statut de l'application
- L'image de l'application
- Son nom
- Sa description courte (synopsis du jeu, nombre max de joueurs, configuration du serveur de jeu, ...)



- Boutons changement de statuts pour une application (seulement disponible pour un administrateur), représenté ici par un bouton POWER pour démarrer ou éteindre l'application, suivi d'un bouton pour redémarrer l'application si elle se trouve déjà en cours de fonctionnement, pour terminer sur un bouton permettant d'activer ou non la supervision d'une application et réagir en fonction de son statut (Ces actions ne sont disponibles seulement pour les administrateurs, elles seront expliquées plus bas dans ce document).

Ici, nous pouvons voir que l'utilisateur est connecté et qu'il peut dès lors accéder à son compte grâce au bouton "Mon compte". Si l'utilisateur clique sur l'image, le nom, la description de l'application, il est directement envoyé vers la page de l'application.

## Détail d'une application



Ce mockup montre le détail d'une application avec sa description complète (synopsis du jeu, nombre max de joueurs, configuration du serveur de jeu, ...) et la possibilité de changer le statut de l'application, si l'utilisateur est connecté en tant qu'administrateur.

On y retrouve l'indicateur indiquant le statut de l'application mais aussi les différents boutons représentant les interactions possibles, cependant disponible seulement pour les administrateurs.

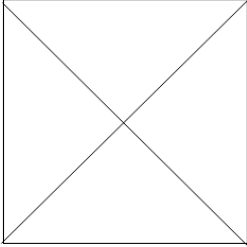
## Liste des suggestions

Menu	The Independent Supervisor																						
	Bienvenue, Mr. X		Mon Compte																				
<ul style="list-style-type: none"> <li>• Page 1</li> <li>• Page 2</li> <li>• Page 3</li> <li>• ...</li> </ul>	<p>Suggestions</p> <table> <tr> <th>Numéro de suggestion</th><th>Username</th><th>Titre</th><th>Action</th></tr> <tr> <td>#012345601</td><td>Mr. X</td><td>...</td><td>Voir Détail</td></tr> <tr> <td>#012345602</td><td>Mr. Y</td><td>...</td><td>Voir Détail</td></tr> <tr> <td>#012345603</td><td>Mme. W</td><td>...</td><td>Voir Détail</td></tr> <tr> <td>#012345604</td><td>Mme. V</td><td>...</td><td>Voir Détail</td></tr> </table>			Numéro de suggestion	Username	Titre	Action	#012345601	Mr. X	...	Voir Détail	#012345602	Mr. Y	...	Voir Détail	#012345603	Mme. W	...	Voir Détail	#012345604	Mme. V	...	Voir Détail
Numéro de suggestion	Username	Titre	Action																				
#012345601	Mr. X	...	Voir Détail																				
#012345602	Mr. Y	...	Voir Détail																				
#012345603	Mme. W	...	Voir Détail																				
#012345604	Mme. V	...	Voir Détail																				

Sur ce mockup, nous pouvons retrouver un tableau listant les différentes suggestions laissée par nos utilisateurs afin de rajouter ou modifier des fonctionnalités ou des applications du parc.

Il est donc possible d'y retrouver le nom d'utilisateur ayant suggéré cette modification, ainsi qu'un titre décrivant en quelques mots la suggestion. A l'instar de notre parc d'application, il est possible de voir plus en détail une suggestion.

## Détail d'une suggestion

Menu	The Independent Supervisor
<ul style="list-style-type: none"> <li>• Page 1</li> <li>• Page 2</li> <li>• Page 3</li> <li>• ...</li> </ul>	<div>Bienvenue, Mr. X <span>Mon Compte</span></div> <div> <p>Suggestion O</p>  <p>           Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque eu porttitor nibh, vel gravida mauris. Vestibulum molestie fermentum nisl. Fusce a eros condimentum, pulvinar metus vitae, hendrerit eros. Pellentesque posuere, tortor nec rutrum cursus, purus magna placerat neque, non tristique metus lorem at ipsum. Donec at odio et risus suscipit maximus id sit amet leo. Quisque vehicula orci sit amet tincidunt euismod. Cras fringilla et nulla nec laoreet. Vivamus rutrum, elit quis pulvinar rhoncus, nibh sem tristique neque, non hendrerit lectus lacus nec metus. Nulla non facilisis ligula, vitae tristique massa. Integer tristique, mi eget sodales placerat, purus turpis iaculis massa, maximus congue velit ipsum rutrum turpis. Sed sagittis pulvinar pharetra. Aenean ac lacinia lectus, accumsan feugiat nisl.         </p> </div>

La page expliquant la suggestion réalisée par un utilisateur sera quasiment identique à celle de la description d'une application, seul l'indicateur et les interactions ne seront pas présent.

Un titre, une description et potentiellement une image suffiront aux administrateurs pour modifier l'application Web ou le parc d'applications pour répondre à la demande de l'utilisateur.

*Les diagrammes de séquences pour les interfaces utilisateurs ci-dessus (« Liste des applications », « Détail d'une application », « Liste des suggestions », « Détail d'une suggestion »), ne se résume qu'à une communication simple entre la partie Front End qui demande les informations à la partie Back End qui elle-même réalise cette demande à la base de données et qu'un retour se fasse afin d'y afficher les données à l'utilisateur sur son navigateur web.*

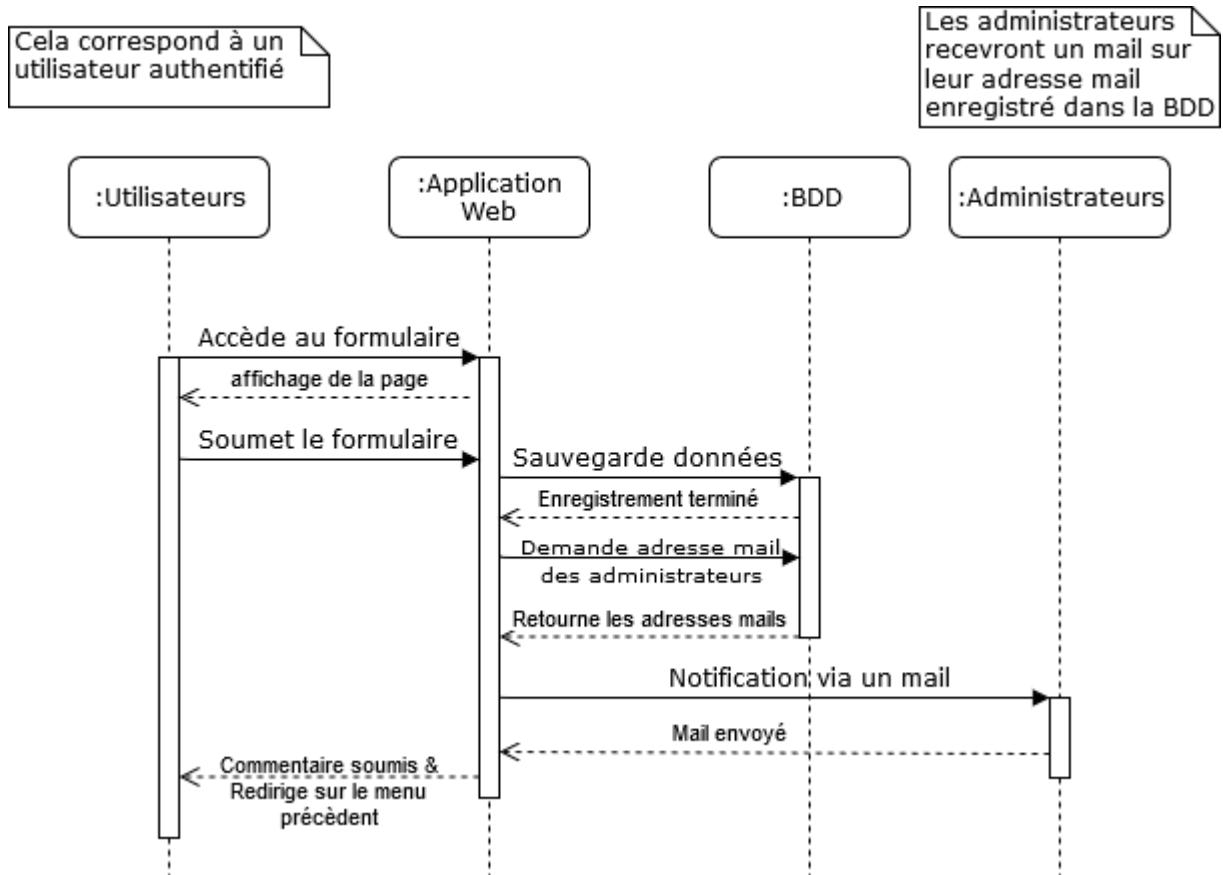
*Ajout d'une idée/ suggestion*

Menu	The Independent Supervisor	Bienvenue, Mr. X	Mon Compte
<ul style="list-style-type: none"><li>• Page 1</li><li>• Page 2</li><li>• Page 3</li><li>• ...</li></ul>	<div>Ajout d'une suggestion</div> <div><div>Titre</div><input type="text"/></div> <div><div>Catégorie</div><input type="text"/></div> <div><div>Description</div><input type="text"/></div> <div><div>Télécharger une image</div><div>Choisir un f...<input type="text"/></div></div> <div><div>Sauvegarder</div><div>Annuler</div></div>		

Dans le but de partager un problème rencontré, ou suggérer une demande de modification sur l'application Web, ainsi que toute demande en lien avec le parc d'applications. Tout utilisateur du site, peut soumettre en remplissant un formulaire disponible depuis la liste des suggestions, en spécifiant un titre décrivant rapidement la suggestion, une catégorie afin de mieux cibler le sujet et une description pour décrire entièrement l'idée que l'utilisateur souhaite partagée.

Pour mieux décrire la suggestion, il est possible de télécharger une image.

Derrière cette interface utilisateur, un traitement est réalisé, ce diagramme de séquence ci-dessous, représente les différentes actions sont réalisés. Notre utilisateur enregistre les différentes informations du formulaire avant de le soumettre. Cette nouvelle suggestion est enregistrée et les administrateurs en sont informés via un communiqué réalisé par un envoi de mail.



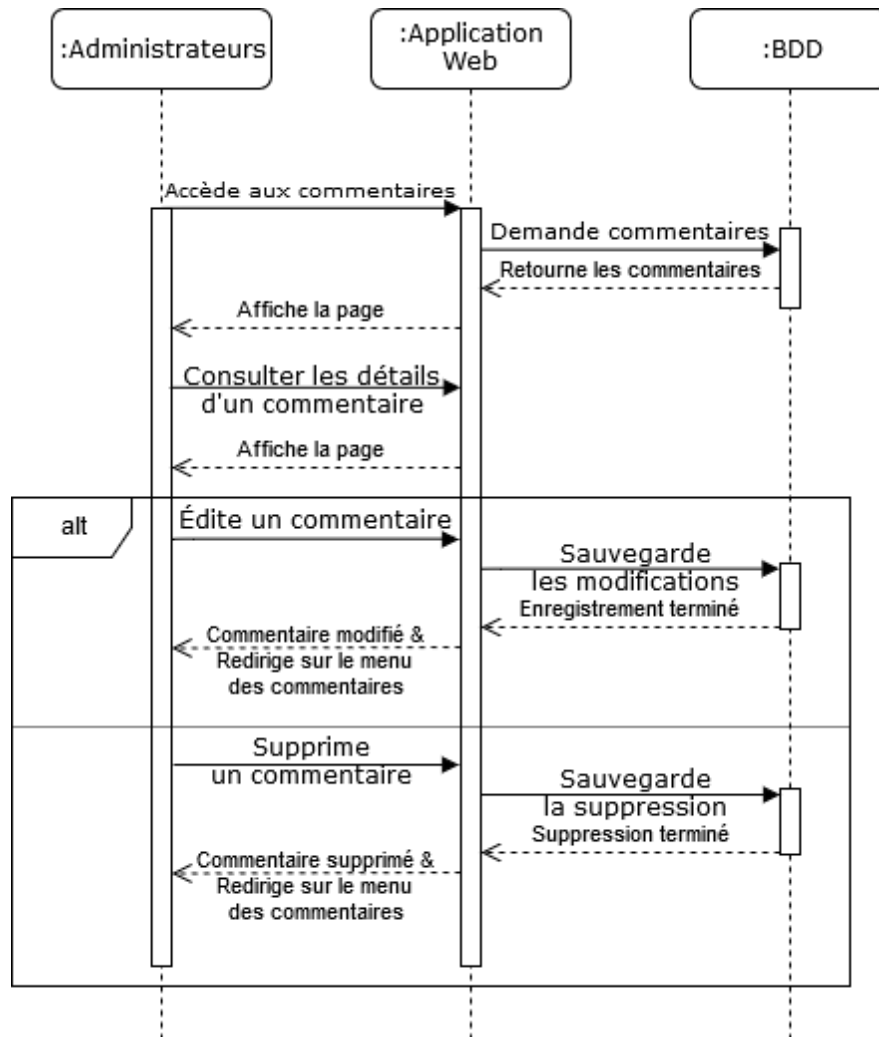
## Édition d'une idée/suggestion

Menu	The Independent Supervisor
<ul style="list-style-type: none"><li>• Page 1</li><li>• Page 2</li><li>• Page 3</li><li>• ...</li></ul>	<div>Bienvenue, Mr. X <a href="#">Mon Compte</a></div> <div><h3>Editer une suggestion</h3><div>Titre <input type="text"/></div><div>Catégorie <input type="text"/></div><div>Description <input type="text"/></div><div>Télécharger une image <input type="text" value="Choisir un f..."/></div><div><a href="#">Sauvegarder</a> <a href="#">Supprimer</a> <a href="#">Annuler</a></div></div>

L'auteur de la suggestion mais aussi les administrateurs pourront éditer une suggestion, pour apporter plus de détails, mais aussi supprimer la suggestion en cas de doublon, ou encore de résolution si une demande de modification a été réalisée, ou dans le cas d'un problème résolu.

Tout comme l'ajout d'une suggestion, le procédé est similaire pour la modification d'une suggestion, la seule modification apparente est qu'il n'y a pas de renvoi de mail indiquant qu'une modification a été réalisé auprès des administrateurs afin d'éviter un envoi de mails trop fréquent.

Il est tout à fait possible de supprimer une suggestion, ce qui en informe le serveur afin que cette modification soit réalisée en base de données et qu'elle ne soit plus apparente depuis le site web. La suppression d'une suggestion n'est pas critique, elle n'est donc pas par défaut archiver.





## Modification des informations utilisateur

Menu	The Independent Supervisor
<ul style="list-style-type: none"> <li>• Page 1</li> <li>• Page 2</li> <li>• Page 3</li> <li>• ...</li> </ul>	<div style="text-align: right;">           Bienvenue, Mr. X <span>Mon Compte</span> </div> <div style="text-align: center; margin-top: 20px;"> <b>Mon compte</b> </div> <div style="margin-top: 10px;"> <div>Nom d'utilisateur</div> <input type="text"/> </div> <div style="margin-top: 5px;"> <div>Mot de passe</div> <input type="password"/> </div> <div style="margin-top: 5px;"> <div>Confirmation</div> <input type="password"/> </div> <div style="margin-top: 5px;"> <div>E-Mail</div> <input type="text"/> </div> <div style="margin-top: 5px;"> <div>Prénom</div> <input type="text"/> </div> <div style="margin-top: 5px;"> <div>Nom</div> <input type="text"/> </div> <div style="margin-top: 10px; display: flex; justify-content: center; gap: 10px;"> <span style="background-color: #d4edda; padding: 5px 10px; border: 1px solid #c3e6cb;">Sauvegarder</span> <span style="background-color: #f8d7da; padding: 5px 10px; border: 1px solid #f5c6cb;">Supprimer son compte</span> </div>

Ce mockup montre comment un utilisateur peut modifier ses informations personnelles ou s'il le souhaite supprimer son compte utilisateur. Il est possible qu'un administrateur puisse accéder à ce formulaire afin d'y modifier, dans le cas d'un mot de passe oublié, ou supprimer le compte d'un utilisateur, si cet utilisateur ne fait plus parti de l'équipe de joueur. Le propriétaire du compte utilisateur aura accès à cette page afin de réaliser ces propres modifications.

*Le diagramme de séquence pour la modification des informations utilisateur peut être assimiler à l'édition d'une suggestion. Ce qui en fait un diagramme quasiment identique sur la forme, seul le contenu des informations qui transite change.*

## b) Administrateur

Pour tous les diagrammes présents ci-dessous, une authentification est nécessaire au préalable, toutes personnes authentifiées, qui possèdent le rôle « Administrateur » auront accès à ces interfaces utilisateurs et pourront réaliser les actions en conséquence.

### *Gestion des utilisateurs*

Menu	The Independent Supervisor																						
	Bienvenue, Mr. X		Mon Compte																				
<ul style="list-style-type: none"> <li>• Page 1</li> <li>• Page 2</li> <li>• Page 3</li> <li>• ...</li> </ul>	<div>Utilisateurs</div> <table> <tr> <th>Numéro d'identifiant</th><th>Username</th><th>Rôle</th><th>Action</th></tr> <tr> <td>#012345601</td><td>Mr. X</td><td>Administrateur</td><td>Voir Détail</td></tr> <tr> <td>#012345602</td><td>Mr. Y</td><td>Utilisateur</td><td>Voir Détail</td></tr> <tr> <td>#012345603</td><td>Mme. W</td><td>Utilisateur</td><td>Voir Détail</td></tr> <tr> <td>#012345604</td><td>Mme. V</td><td>Administrateur</td><td>Voir Détail</td></tr> </table>			Numéro d'identifiant	Username	Rôle	Action	#012345601	Mr. X	Administrateur	Voir Détail	#012345602	Mr. Y	Utilisateur	Voir Détail	#012345603	Mme. W	Utilisateur	Voir Détail	#012345604	Mme. V	Administrateur	Voir Détail
Numéro d'identifiant	Username	Rôle	Action																				
#012345601	Mr. X	Administrateur	Voir Détail																				
#012345602	Mr. Y	Utilisateur	Voir Détail																				
#012345603	Mme. W	Utilisateur	Voir Détail																				
#012345604	Mme. V	Administrateur	Voir Détail																				

Un administrateur peut accéder à cette page listant les utilisateurs de l'application, représenté par ce mockup. Ainsi il pourra éditer un utilisateur afin d'en changer les informations liées.

*Le diagramme de séquence pour obtenir la liste des utilisateurs ne se résume qu'à une communication simple entre la partie Front End qui demande les informations à la partie Back End qui elle-même réalise cette demande à la base de données et qu'un retour se fasse afin d'y afficher les données à l'administrateur sur son navigateur web.*

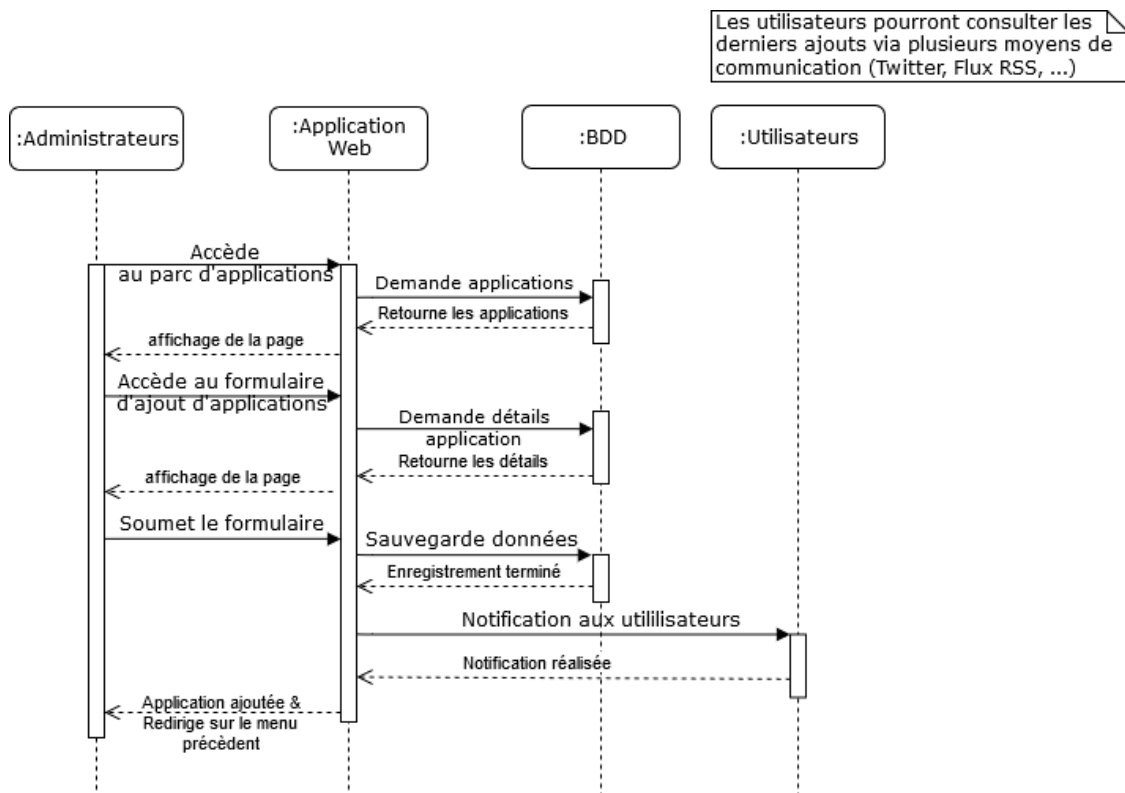
## Ajout d'une application

Menu	The Independent Supervisor	Bienvenue, Mr. X	Mon Compte
<ul style="list-style-type: none"><li>• Page 1</li><li>• Page 2</li><li>• Page 3</li><li>• ...</li></ul>	<div><h3>Ajout d'une application</h3><div><div>Nom de l'application</div><input type="text"/></div><div><div>Description</div><input type="text"/></div><div><div>Fichier de configuration</div><input type="text"/></div><div><div>Commande pour le démarrage</div><input type="text"/></div><div><div>Commande pour la mise à jour</div><input type="text"/></div><div><div>Télécharger une image</div><div>Choisir un f...<input type="text"/></div></div></div> <div><div>Sauvegarder</div><div>Annuler</div></div>		

Pour ajouter une application, un administrateur pourra accéder à cette page depuis la page listant les applications. Il pourra spécifier le nom de l'application, une description, les chemins vers les scripts (avec leur paramètre si besoin) permettant de démarrer et mettre à jour l'application, pour accompagner le tout, il sera possible d'ajouter une image à la fiche de la nouvelle application.

Les informations pourront changer en fonction des besoins et donc être ainsi modulable pour s'adapter à chaque type d'application.

L'ajout d'une application est très similaire à celle d'une suggestion, ce qui en fait des diagrammes de séquence très ressemblant. Les seules différences sont présentes autour des informations qui transite mais aussi à la communication réalisée en fin d'ajout. Au lieu de réalisé une communication seulement aux administrateurs, cette fois ci, le communiqué cible tout utilisateurs du site web.

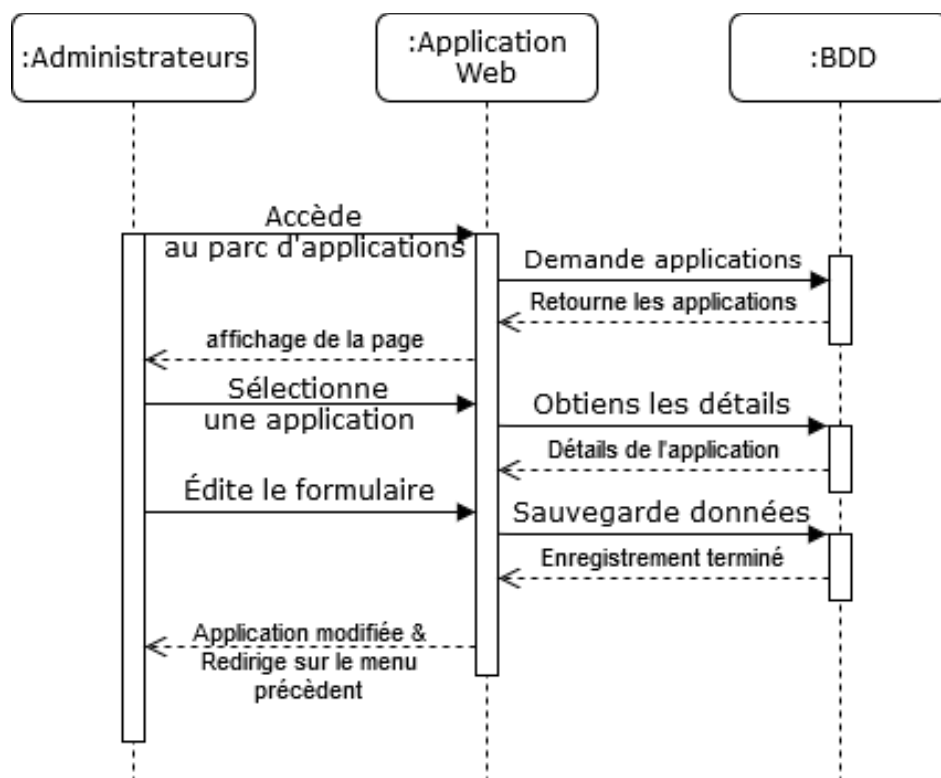


### Modification d'une configuration d'une application

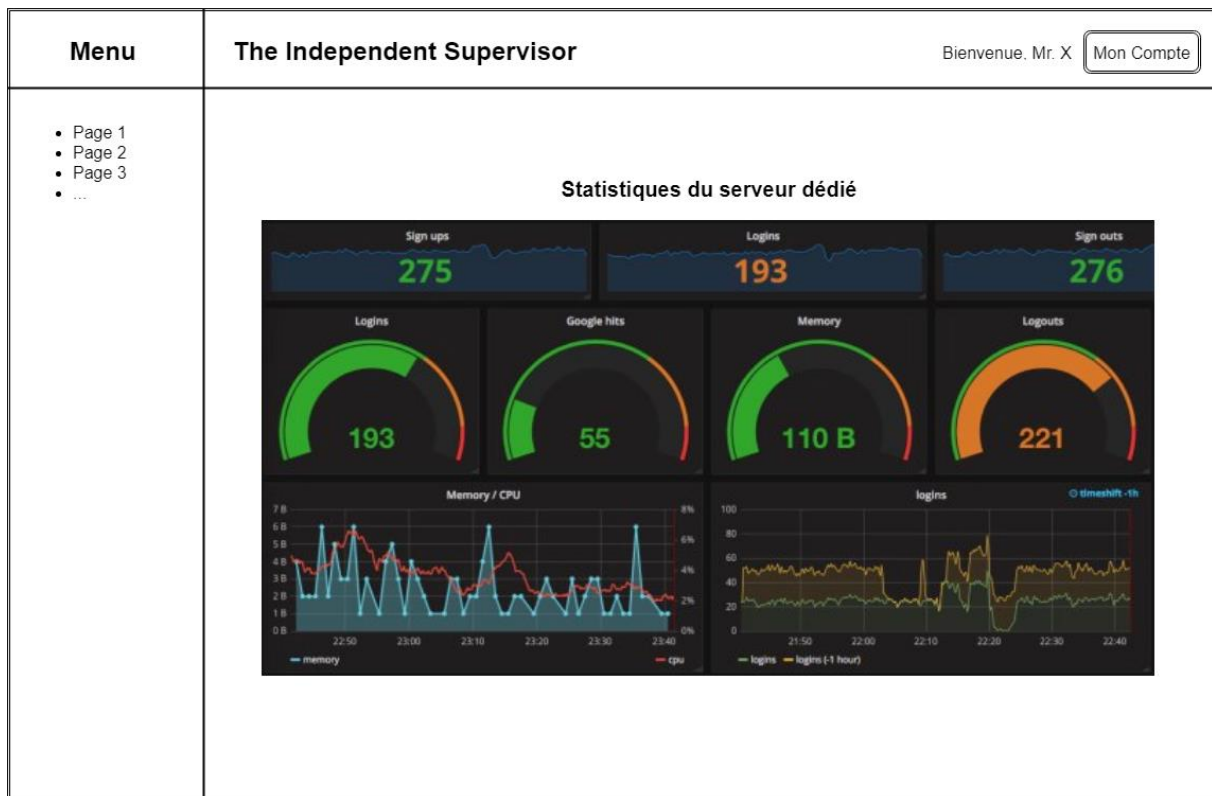
Menu	The Independent Supervisor	Bienvenue, Mr. X	Mon Compte
<ul style="list-style-type: none"><li>• Page 1</li><li>• Page 2</li><li>• Page 3</li><li>• ...</li></ul>	<div><h3>Editer une application</h3><div><div>Nom de l'application</div><input type="text"/></div><div><div>Description</div><input type="text"/></div><div><div>Fichier de configuration</div><input type="text"/></div><div><div>Commande pour le démarrage</div><input type="text"/></div><div><div>Commande pour la mise à jour</div><input type="text"/></div><div><div>Télécharger une image</div><div>Choisir un f...<input type="text"/></div></div></div> <div><div>Sauvegarder</div><div>Supprimer</div><div>Annuler</div></div>		

Tout comme l'ajout d'une application, l'administrateur pourra accéder à cette page depuis la liste du parc d'applications. Les informations enregistrées sont aussi identiques. L'administrateur pourra aussi supprimer une application qui ne fait plus parti du parc d'application.

Modifier ou supprimer une application ne change pas d'une modification ou suppression d'une suggestion exceptée encore une fois des informations qui transite.



## Consultation des ressources serveurs



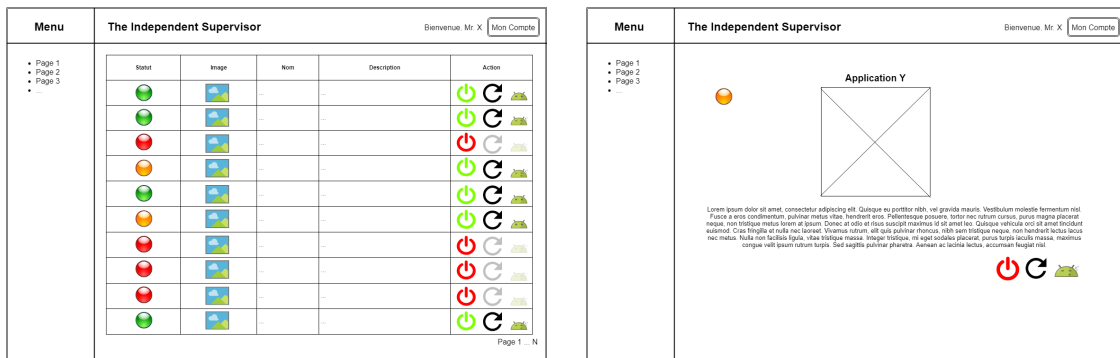
Pour que les administrateurs puissent se rendre compte de la consommation des applications, une page composée de plusieurs indicateurs sera présente. Il sera donc possible par exemple de suivre l'espace de stockage disponible, le taux d'utilisateur du processeur, ou encore de la mémoire vive.

Il n'est pas impossible que d'autres indicateurs viennent à s'ajouter au fil du développement, lié au parc d'applications, ou à l'application web ou encore au serveur dédié.

Ces indicateurs ne possèdent qu'une valeur indicative pour les administrateurs, tous ne sont pas qualifié pour en déterminer une quelconque action à réaliser. Des actions peuvent être entrepris par un administrateur plus qualifié ou par le superviseur qui détecte automatiquement s'il y a un dépassement d'un seuil. Par exemple, une consommation trop importante de la mémoire vive, suite à cela, après une détermination par priorité, des applications peuvent être sélectionner pour être éteinte et ainsi réduire cette consommation.

Ces informations pourront être archivés pendant plusieurs afin de récupérer un historique de consommation et réaliser différentes analyses sur la consommation de ressources.

### c) Interactions avec le parc d'application

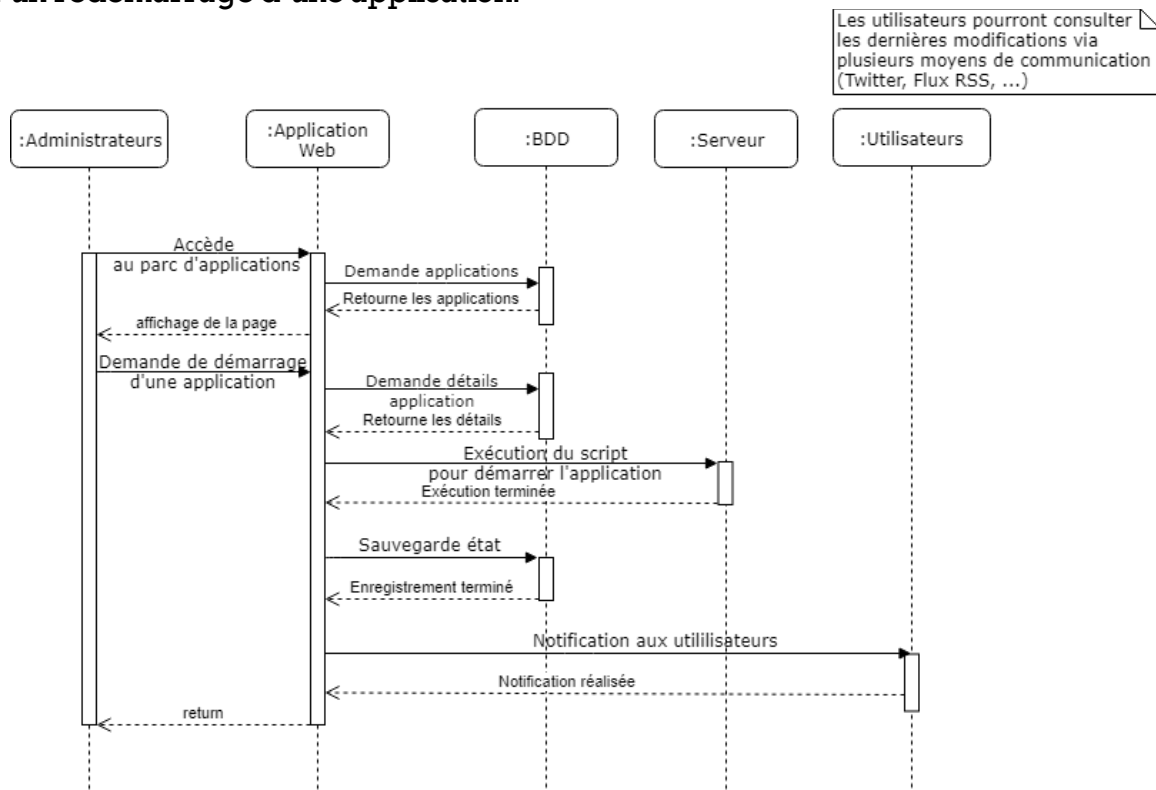


Les différentes interactions disponibles sur les interfaces utilisateurs ont des impacts différents sur le parc d'applications. Ces interactions sont disponibles par exemple sur la liste des applications ou en regardant plus en détail une application.

Il y a cependant des interactions qui sont plus transparentes pour les administrateurs du site web, effectivement l'arrêt d'une application suite à une consommation trop importante de ressources n'est pas réalisé par un administrateur mais bel et bien par le superviseur qui se charge du suivi de la consommation du parc d'applications.

#### Administrateur Démarrer une application

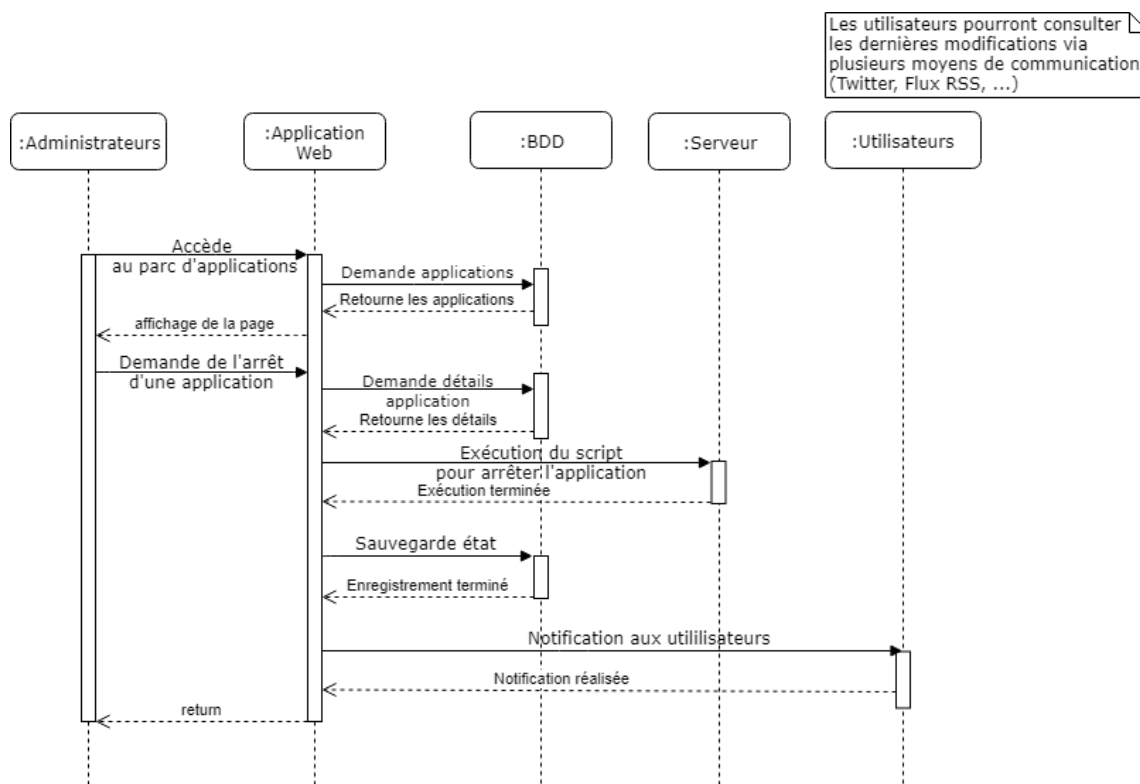
Il sera possible de réaliser une mise à jour automatique lors d'un démarrage ou d'un redémarrage d'une application.





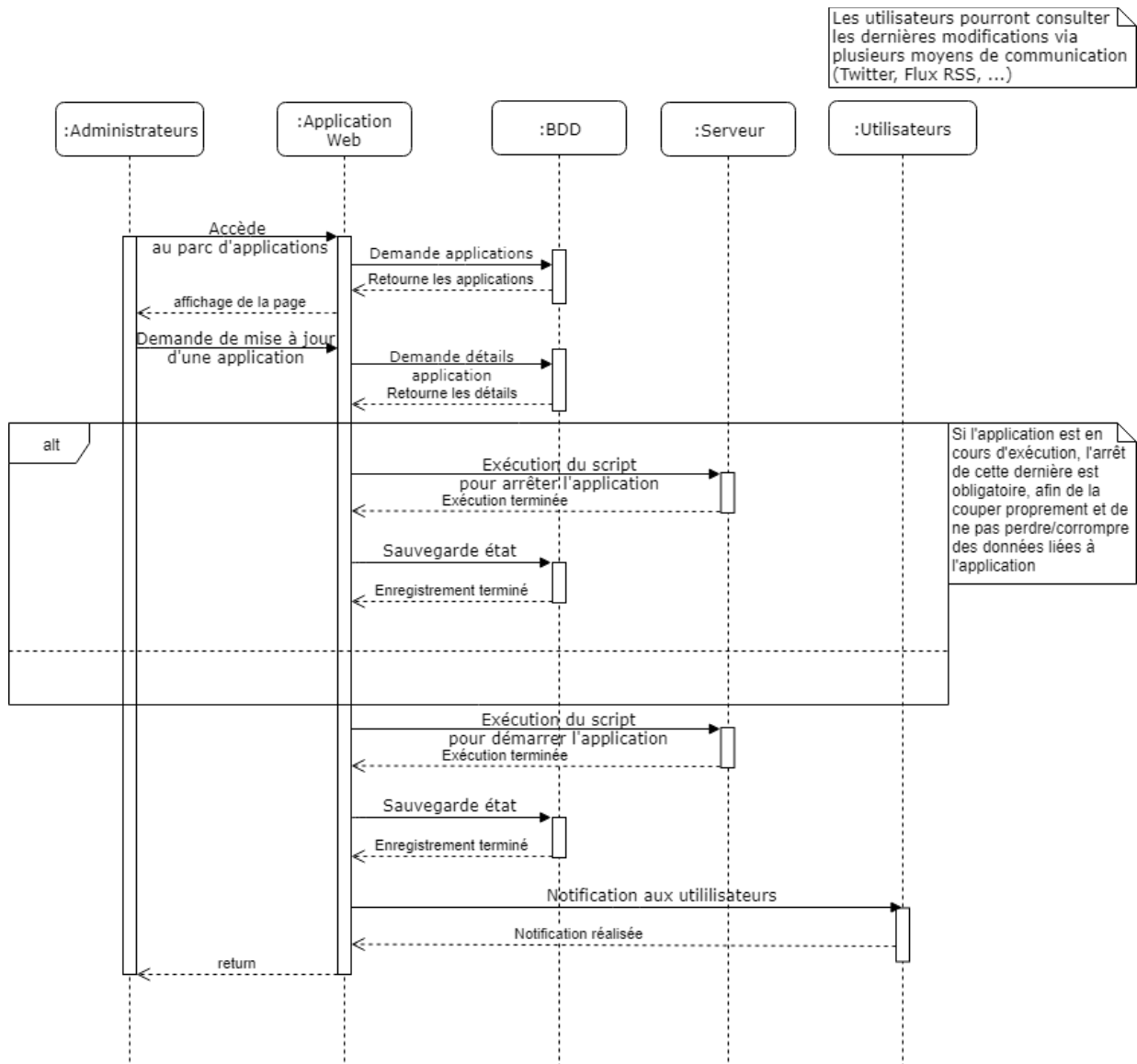
## Arrêter une application

L'arrêt d'une application ne se fait pas brutalement, un message est envoyé via un canal de communication, la plupart des applications qui sont la cible de ce projet en possèdent, elles permettent notamment d'écrire des commandes spécifiques et dans la plupart des cas, elles permettent d'envoyer un message qui sera visible pour tous les utilisateurs étant connecté au serveur. Ainsi, il est possible de prévenir les utilisateurs d'un arrêt imminent et de leur laisser un temps déterminé pour qu'il puisse terminer ce qu'ils étaient en train de réaliser. Cette communication sera présente au sein du script qui sera exécuté.



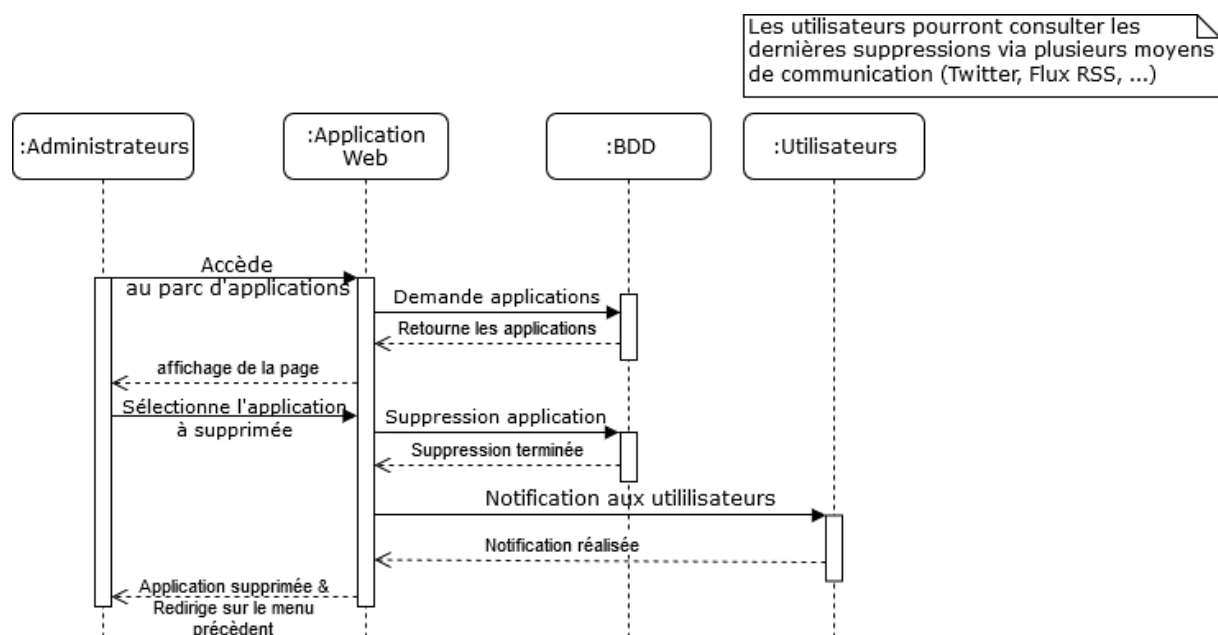
## Mise à jour d'une application

Tout comme l'arrêt d'une application, pour la mise à jour d'une application et seulement si l'application est en cours d'exécution, un message sera envoyé de la même manière pour prévenir cet arrêt temporaire pour réaliser une mise à jour de l'application en question.



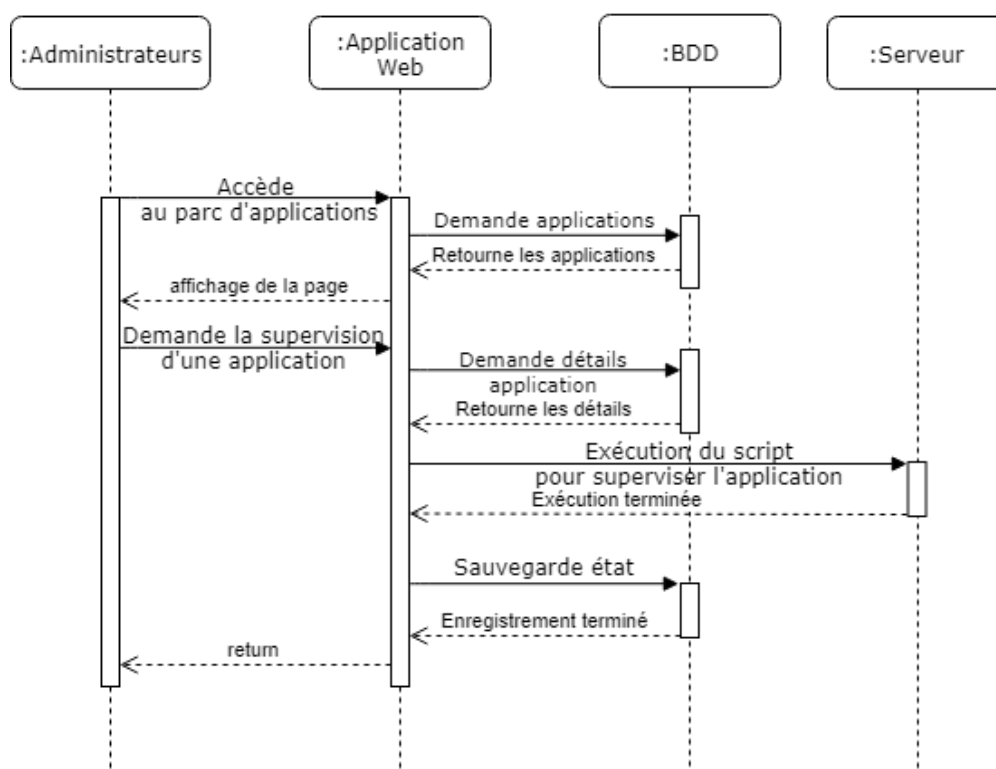
## Suppression d'une application

La suppression d'une application pourra être disponible depuis la liste des applications, le détail d'une application ou moment de l'édition d'une application.



## Supervision d'une application

En activant la supervision d'une application, cela permet au superviseur d'avoir des interactions avec l'application et ainsi réaliser des actions automatiques.

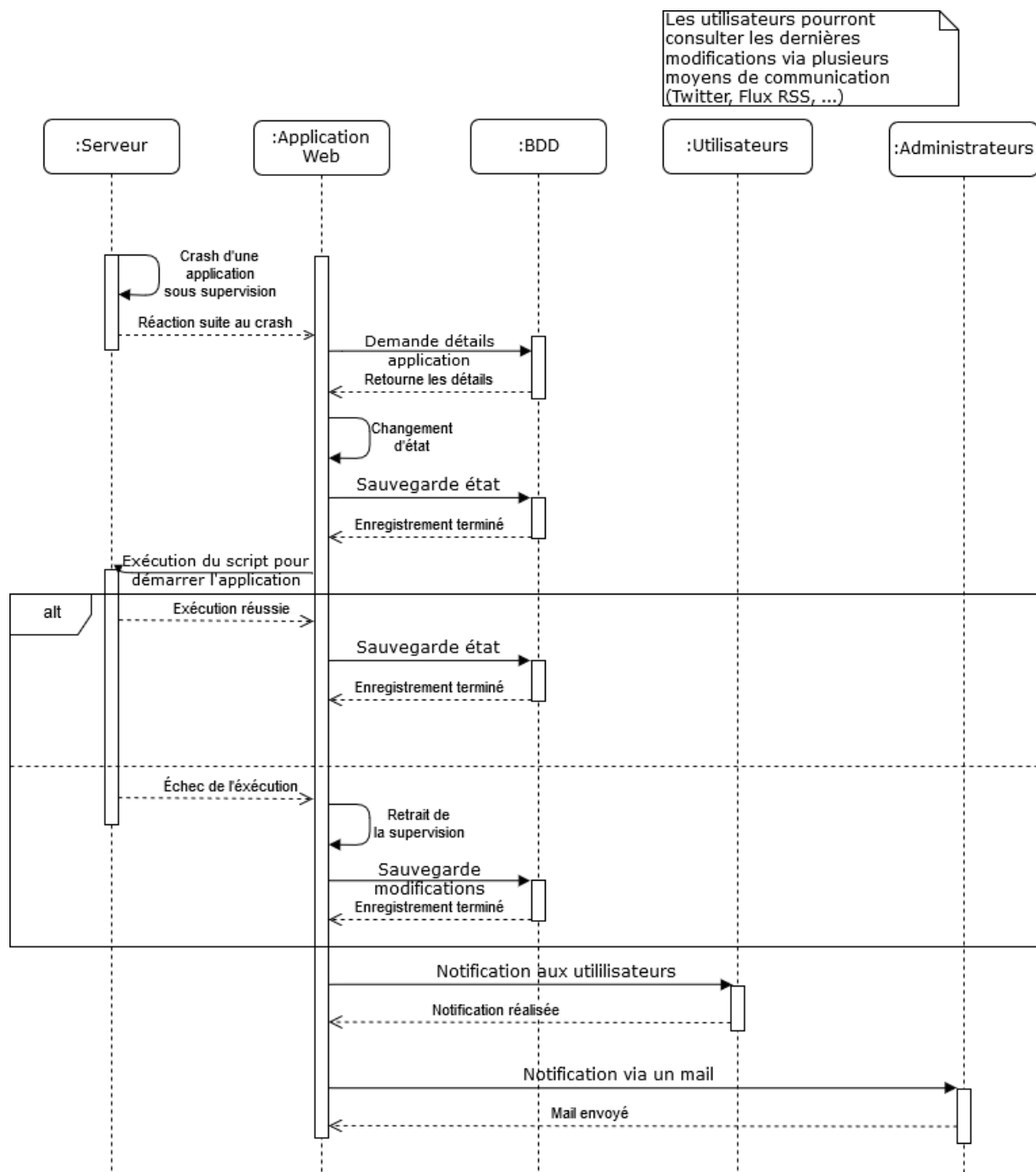


## Superviseur

### Redémarrage automatique en cas de crash d'une application

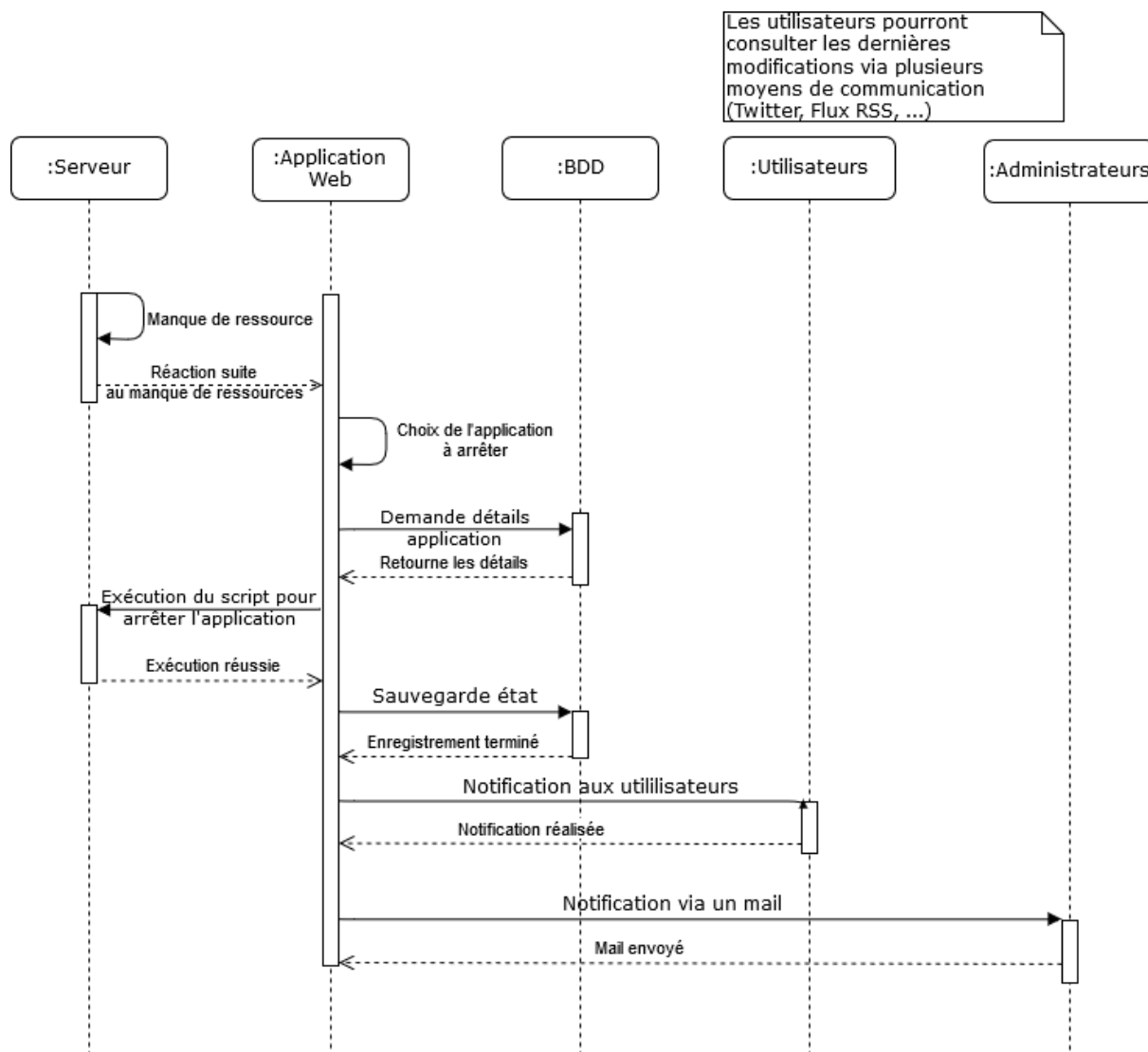
Il est possible qu'une application cesse de fonctionner pour une raison X ou Y, cependant si cette application est sous supervision, il est possible qu'elle doit être redémarrer automatiquement, ce qui fait qu'on change les informations autour du statut de l'application puis on réalise une tentative de redémarrage. Si cela ne fonctionne pas à nouveau, les administrateurs en sont prévenus du problème rencontré lors du redémarrage. Les utilisateurs quant à eux sont prévenus qu'une application est passer à l'arrêt.

Une mise à jour automatique de l'application peut être réalisé lors d'un redémarrage.

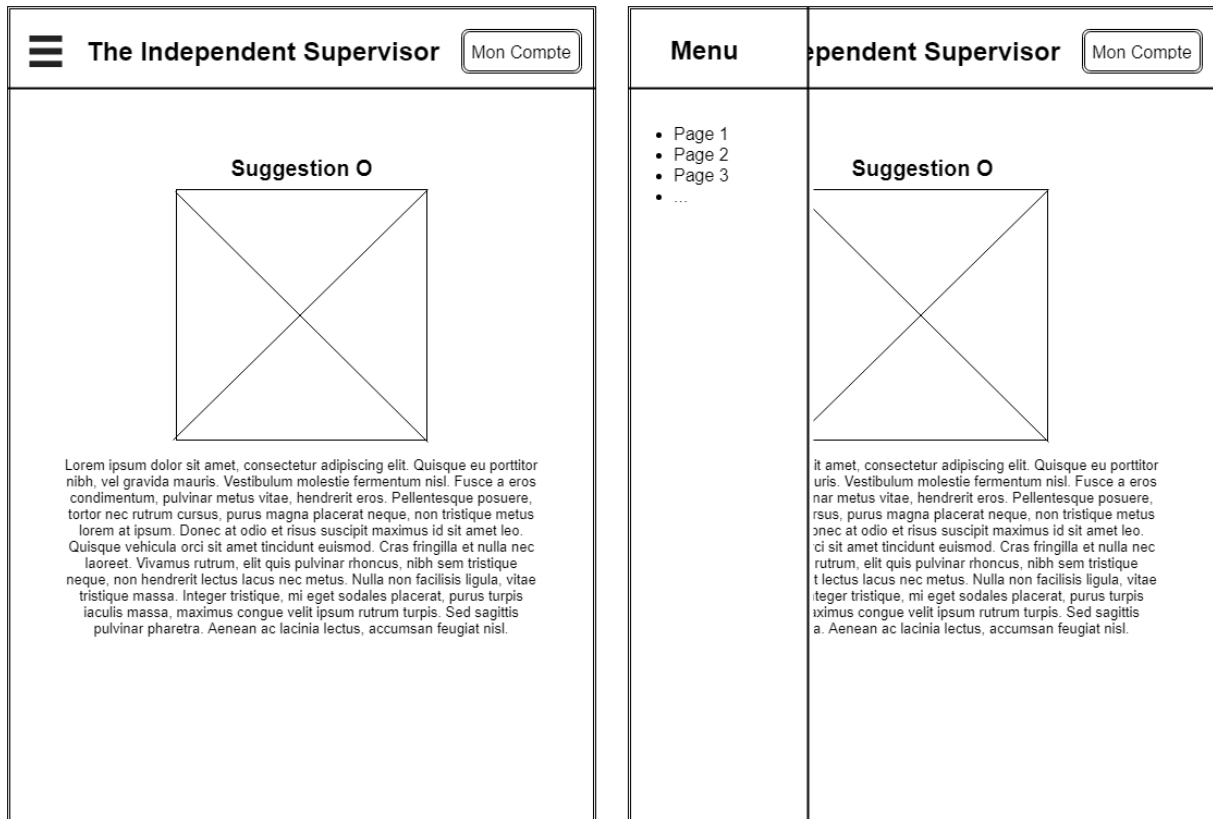


## Arrêt automatique en cas de manque de ressources

Une application peut cesser de fonctionner pour une raison quelconque, mais aussi cela peut arriver que ça soit le serveur qui choisisse de la couper. Il est donc important d'éviter de rencontrer ce genre de cas, c'est pourquoi en réalisant une supervision sur une application, on peut réaliser une action préventive en gardant un suivi de la consommation des ressources du serveur. Si une consommation devient trop importante, une ou plusieurs applications doivent être interrompu correctement (tel qu'un arrêt réalisé par un administrateur) afin de réduire cette charge.

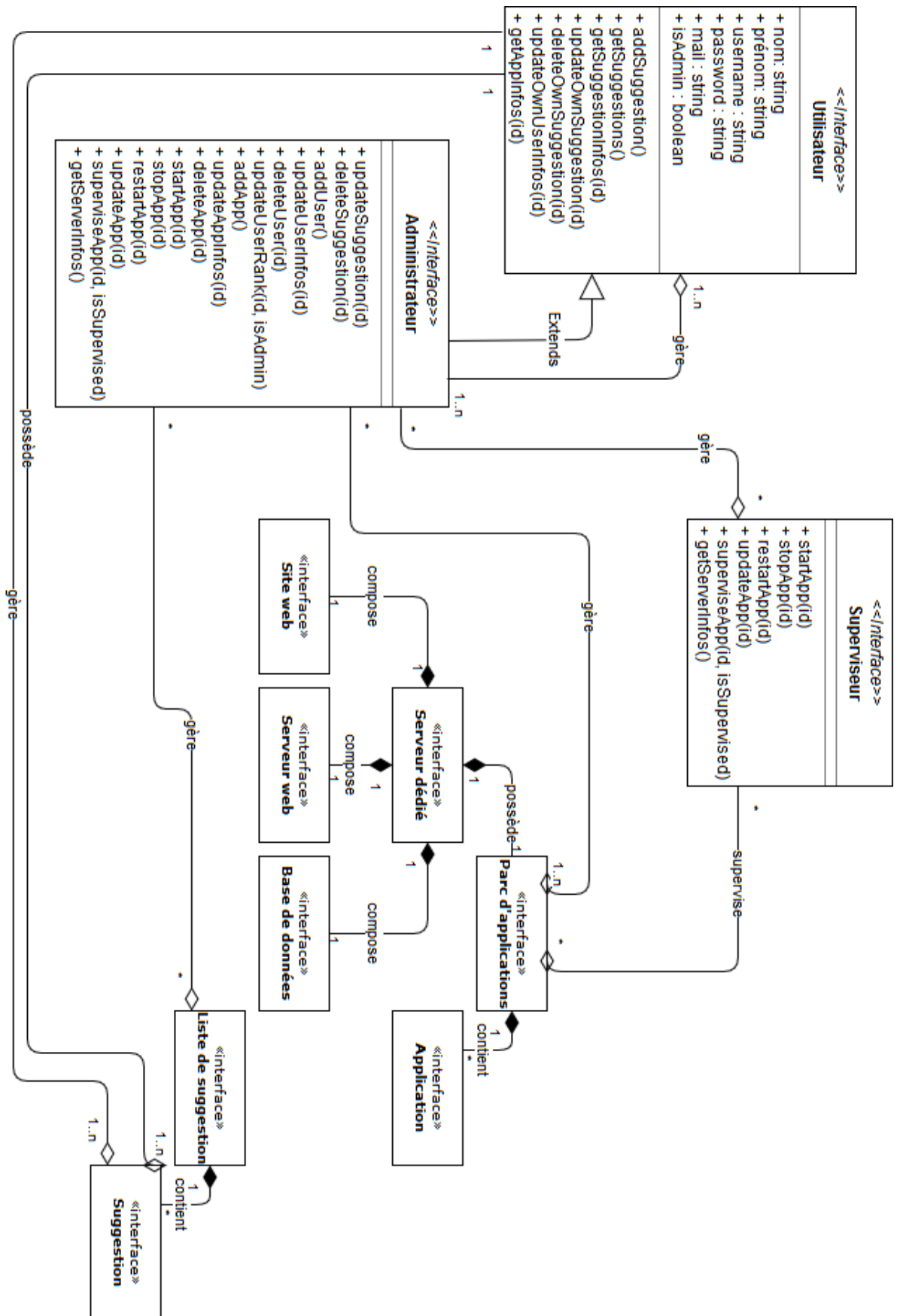


## d) Vue mobile



L'application web doit être aussi disponible sur mobile, il est donc important que le contenu s'adapte en fonction de la résolution. Pour cela le contenu du menu habituellement directement disponible sur la page est caché pour la visualisation depuis un smartphone. Pour le rendre visible, il faut cliquer sur le bouton présent en haut à gauche de la page, tout le reste de la page passe en arrière-plan, afin que l'utilisateur puisse interagir avec le menu.

## e) Diagramme de classes



## 2. MEAN Stack



MEAN est un paquet de logiciel JavaScript gratuit et open-source pour la création de sites Web dynamiques et d'applications Web.

Le paquet MEAN est composé de MongoDB, Express.js, Angular et Node.js. Comme tous les composants des programmes de support de la pile MEAN sont écrits en JavaScript, les applications MEAN peuvent être écrites dans un seul langage pour les environnements d'exécution côté serveur et côté client (cependant côté client, le langage TypeScript sera utilisé, contrairement avec une utilisation d'AngularJS, le langage JavaScript serait utilisé).

L'acronyme MEAN a été inventé par Valeri Karpov, un développeur de MongoDB. Les composants de la pile MEAN sont les suivants :

- MongoDB, base de données NoSQL
- Express.js, framework web tournant sur Node.js
- Angular, (communément appelé "Angular 2+" ou "Angular v2 et plus") est un framework d'application web open-source basé sur TypeScript.
- Node.js, un environnement d'exécution pour les applications serveur et réseau orientées événements.



## a) MongoDB



### *Base de données*

Une base de données est une collection d'informations organisées afin d'être facilement consultables, gérables et mises à jour facilement. Au sein d'une base de données, les données sont organisées en lignes, colonnes et tableaux. Elles sont indexées afin de pouvoir facilement trouver les informations recherchées à l'aide d'un logiciel informatique. À chaque modification de la base de données de nouvelles informations sont ajoutées ou sont mises à jour, voir éventuellement supprimées.

Elles peuvent se charger de créer, de mettre à jour ou de supprimer les données souhaitées. Elles peuvent permettre d'effectuer des recherches parmi les données qu'elles contiennent sur demande de l'utilisateur.

Les bases de données sont stockées sous forme de fichiers ou d'ensemble de fichiers sur tout type d'appareil de stockage. Les bases de données dites traditionnelles sont organisées par champs, enregistrements et fichiers. Un champ est une seule pièce d'information (ex : ville => TOURS). Un enregistrement est un ensemble de champs (ex : prénom => Brandon, nom => SIMON-VERMOT, ville => TOURS, ...). Ces enregistrements sont stockés sous formes de fichiers.

Comme parfait exemple, un répertoire téléphonique peut représenter un fichier qui compose la base de données. Ce répertoire est constitué d'un ensemble d'enregistrements, dont chaque enregistrement regroupe trois champs : nom, adresse et numéro de téléphone.

## Orienté documents

MongoDB est un système de base de données orienté documents. À l'inverse avec un système de base de données « traditionnel », dites base de données relationnelle, les données sont stockées par ligne dans des tables. Et il est souvent nécessaire de faire des jointures sur plusieurs tables afin de tirer des informations assez pertinentes de la base.

Dans un système orienté documents, les données sont modélisées sous forme de document sous un format JSON.

On ne parle plus de tables et d'enregistrements mais de collections et de documents. Ce système de gestion de données nous évite ainsi de faire des jointures de tables car toutes les informations propres à un certain donnée sont stockées dans un même document.

Les documents sont les unités de base dans une base MongoDB. Ils sont équivalents aux objets JSON et sont comparables aux enregistrements d'une table dans une base de données relationnelle.

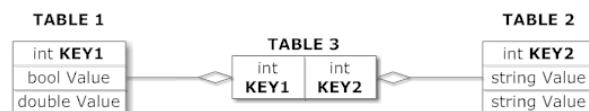
Tout document appartient à une collection et a un champ appelé `_id` qui identifie le document dans la base de données. Le champ `_id` est donc unique afin d'identifier correctement un document.

MongoDB enregistre les documents sur l'espace de stockage sous un format BSON (JSON binaire).

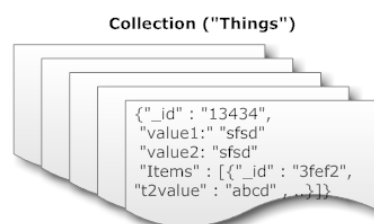
Une collection est un ensemble de documents, l'équivalent d'une table en relationnel. Contrairement aux bases de données relationnelles, les champs des documents d'une collection sont libres et peuvent être différents d'un document à un autre. Le seul champ commun est obligatoire est le champ `"_id"`.

Néanmoins pour que la base soit maintenable, il est préférable d'avoir dans une collection des documents de même type.

### Relational Model



### Document Model



## b) Angular



Le framework Angular 2 (successeur de AngularJS de Google) est un framework JavaScript pour créer des applications monopages (SPA : Single Page Application), web et mobiles.

Grâce à ce framework, il est possible de réaliser différents projets. On peut développer des petits outils interactifs pour un site web existant (moteur de recherche, module de réservation).

On peut constituer un site web complet, qui peut être totalement compatible sur un navigateur web mobile mais aussi le développement d'une application mobile est disponible.

Angular2 est un framework orienté composants. Lors d'un développement d'une application, en réalité une multitude de petits composants sont codés, qui une fois assemblés tous ensemble, constitueront une application à part entière. Un composant est l'assemblage d'un morceau de code HTML, et d'une classe Javascript dédiés à une tâche particulière.

Ces composants reposent sur le standard des Web Components, pensés pour découper une page web en fonction de leur rôle (barre de navigation, boîte de dialogue pour chatter, contenu principal d'une page). Un composant est censé être une partie qui fonctionne de manière autonome dans une application.

Toute une application tiendra dans une simple page HTML, Angular fournit plusieurs librairies, dont certaines d'entre elles forment le coeur du framework.

Les 4 éléments à la base de toute application Angular 2 :

- Module
- Composant
- Template
- Métadonnées

## Module

Un module est une brique d'une application, qui, une fois assemblée à d'autres briques, forme une application à part entière. Chaque module étant dédié à une fonctionnalité particulière.

Par exemple, le module `@angular/core` est un module Angular 2 qui contient la plupart des éléments de base dont nous aurons besoin pour construire une application.

## Composant

Des composants sont la base des applications Angular 2, chaque composant contrôlera un bout de l'interface utilisateur, aussi appelé vue.

La logique d'application d'un composant (ce qu'il doit faire pour supporter la vue) est défini à l'intérieur d'une classe, et cette classe interagit avec la vue à travers un ensemble de propriétés et de méthodes.

Chaque nom de composant est suffixé par "Component", afin de respecter un standard.

## Template

Un template est simplement une "vue", associée à un composant. A chaque fois que l'on définit un composant, on lui associe toujours un template. Un template est une forme spéciale de HTML qui dit à Angular ce que doit afficher le composant.

Parfois un template contient simplement des balises HTML, ce qui n'a rien de particulier, mais il est possible de rendre dynamique ce template en insérant des variables qui pourront être modifiées au cours du visionnage de la page, sans pour autant recharger cette page.

Angular identifie une variable grâce à une syntaxe particulière, le nom de la variable doit être entouré d'accolades, on indique ainsi à Angular 2 que cette variable n'est pas disponible dans le template mais que la valeur de cette variable se trouve dans le composant qui gère ce template.

## Métadonnées

Les métadonnées indiquent à Angular comment traiter une classe. Une classe peut très bien avoir le suffixe 'Component' pour préciser à la lecture que c'est une classe de composant.

Cependant comment Angular sait qu'il s'agit vraiment d'un composant et pas d'un modèle, ou d'un service ?

Grâce aux métadonnées nous pouvons ajouter l'annotation `@component` pour indiquer à Angular que cette classe est un composant.

Les annotations ont souvent besoin d'un paramètre de configuration. Le décorateur `@component` n'échappe pas à cette règle, et prend en paramètre un objet qui contient les informations dont Angular a besoin pour créer et lier le composant et son template.

Les deux options de configuration obligatoires lors de la définition d'un composant sont :

- **Selector** : Un sélecteur est un identifiant unique dans une application, qui indique à Angular de créer et d'insérer une instance de ce composant à chaque fois qu'il trouve une balise `<my-app></my-app>` dans du code HTML d'un composant parent.
- **Template** : Le code du template lui-même, il est également possible d'écrire le code du template dans un fichier à part, auquel cas, il faudra remplacer `template` par `templateURL` et indiquer le chemin relatif vers le fichier du template.

### c) Node.JS



Node.js est un environnement permettant d'exécuter du code JavaScript hors d'un navigateur.

Son architecture est modulaire et événementielle. Il est fortement orienté réseau en possédant pour les principaux systèmes d'exploitation (Unix/Linux, Windows, macOS) de nombreux modules réseau (dont voici les principaux par ordre alphabétique : DNS, HTTP, TCP, TLS/SSL, UDP). De ce fait, il remplace avantageusement, un serveur web tel qu'Apache.

Créé par Ryan Lienhart Dahl en 2009, cet environnement est devenu rapidement très populaire pour ses deux qualités principales :

- Sa légèreté (en corollaire de sa modularité).
- Son efficacité induite par son architecture monothread (en corollaire de la gestion événementielle que propose nativement l'environnement JavaScript).

Intégrer Node.js dans le développement d'applications web participe donc à la logique actuelle de rendre les opérations d'accès aux données les moins bloquantes possible (pour dépasser la problématique dite du « bound I/O » selon laquelle, avant toute autre cause, la latence globale d'une application est due au temps de latence des accès aux données).

Node.js permet donc, pour les applications web, de créer des serveurs extrêmement réactifs.

#### d) Express



Express est un framework web minimaliste, simple mais néanmoins extrêmement puissant. Le squelette d'application qui est généré permet de créer rapidement un programme qui fonctionne.

Ce framework étant basé sur le concept de middlewares, c'est-à-dire des sortent de plugin ou librairie qui propose un service spécifique. Par exemple dans la spécificité d'un web service nous pourrions avoir à faire à la récupération de donnée Json.

### e) Pourquoi ce choix ?

L'utilisation de la pile MEAN apporte d'énormes avantages et permet de gagner un temps sur la prise en main des technologies, puisqu'il n'y a qu'un seul langage à connaître pour pouvoir utiliser ces quatre technologies, Angular 2 utilise du TypeScript, mais cela reste très proche.

Node.JS est à un statut « Production Ready » ce qui veut dire que l'on peut l'utiliser de manière sereine, sans risque de bugs trop importants et que les performances sont présentes, même pour un projet conséquent.

Les performances sont justifiées par le fait qu'il n'y ait pas d'intermédiaire, le code est compilé directement en langage machine, ce qui l'application très performante. Être non-bloquant, c'est le concept de Node.JS, de cette manière, les tâches sont réparties correctement, ce qui ne signifie pas de blocage même pour une tâche longue, ainsi qu'il n'y pas de grosse charge mémoire.

La communauté de Node.JS est présente pour justifier la véracité de cet engouement pour cet outil. Beaucoup de tutoriels, de paquets, articles sont présents afin de répondre à presque tous les problèmes.

L'utilisation du framework Express, permet en plus d'avoir un outil performant côté serveur via Node.JS, un outil qui simplifie la création d'un serveur web. Il permet donc de se concentrer sur la partie logique et non sur les rouages bas niveau. Cet outil permet aussi de réduire la complexité pour sécuriser une API, et donc créer des routes qui nécessitent une authentification.

Une base de données NoSQL, telle que MongoDB, s'harmonise avec la pile MEAN, car la communication est simplifiée puisque la plupart des communications sont réalisées dans un format JSON, ce qui n'épargne pas les requêtes visant la base de données.

MongoDB est certes bien plus performant qu'une base de données SQL dans certains cas, un module rajoute aussi de la simplicité d'utilisation, Mongoose.js, il permet de rajouter beaucoup de fonctionnalités, notamment la possibilité de créer un schéma (dans le cas d'un profil : nom, prénom, ville, numéro, adresse) qui par la suite permettra de créer un module basé sur ce schéma et ainsi créer une régularité dans l'architecture des données.

Angular quant à lui permet de créer une application monopage, seulement une page est chargée pour l'utilisateur puis plusieurs composants vont se rajouter et se mettre à jour sans que l'utilisateur perçoive un rechargement de la page. Cette application peut s'adapter facilement à différent format, différentes résolutions, ce qui permet de créer une application web disponible sur tous types de support.

L'utilisation de la pile MEAN permet d'obtenir une application web multiplateforme rapidement, sans avoir le besoin d'apprendre différents langages, montrant la nécessité d'avoir plusieurs experts côté Front End et Back End, tout cela accompagné d'une communauté forte pour assurer une bonne maintenabilité pendant encore de longues années, ce qui respecte les besoins du projet.



### 3. Serveur dédié

#### a) Hébergement

*Online.net*



Online.net fournit des services d'hébergement pour toutes les tailles d'acteurs Internet dans le monde entier. Ils ont commencé en 1999 et hébergent plusieurs centaines de milliers de sites Internet sur nos trois datacenters. Online.net propose des services de noms de domaine, d'hébergement mutualisé, de serveurs dédiés et d'hébergement en datacenter.

Ils sont à ce jour l'un des leaders de l'hébergement Internet en France.

Online est l'un des acteurs majeurs de l'hébergement en France. Créé en 2000 en même temps que le service d'accès à Internet Free, ils sont une filiale à 100% du groupe Iliad, focalisée sur les services d'hébergement Internet destinés aux webmasters et professionnels de l'Internet.

Un serveur dédié est fourni par le client, hébergé par Online.net, pour le déploiement de l'application et les différents tests. Celui-ci affiche un taux de disponibilité de 99,95%.

#### *DNS*

Afin de ne divulguer l'adresse IP en clair, il est intéressant de passer par un service DNS, afin de référencer le serveur dédié par un nom de domaine (www.exemple.com).

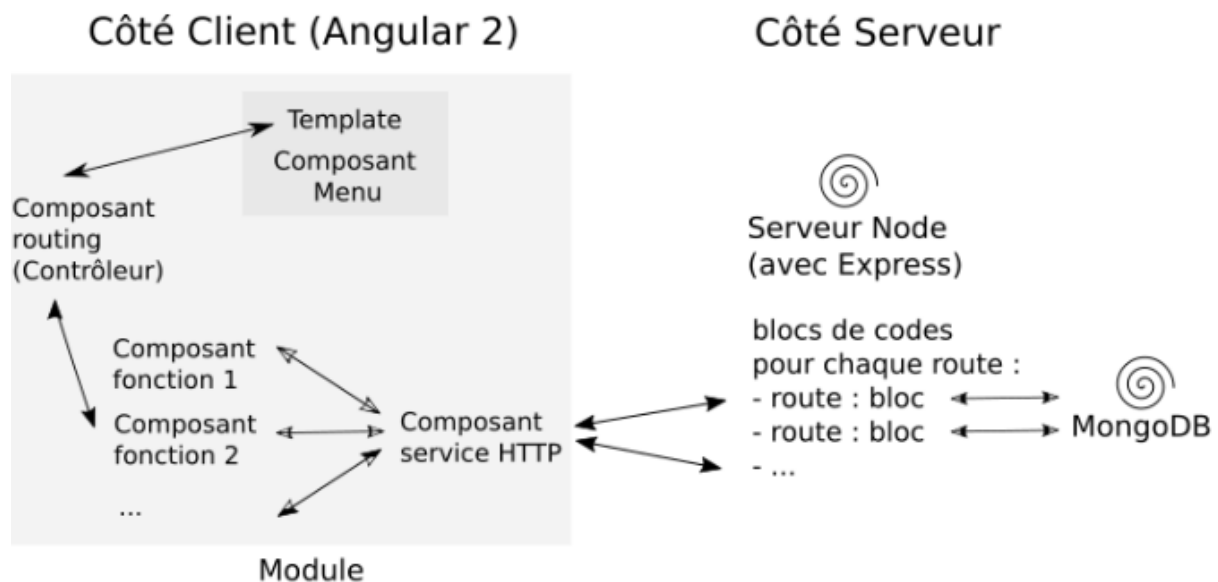
Plusieurs services sont disponibles en recherchant sur Internet, il est possible même d'utiliser un service gratuit dans un premier temps. Un passage vers un service payant sera à la charge du client, il pourra donc décider de passer ou non vers un service payant. Il pourra ainsi choisir Online.net pour centraliser tout via le même hébergeur ou tout autre service.

## b) Architecture

### *Serveur Web (Front End & Back End)*

Le graphique ci-dessous présente l'architecture d'une application web MEAN :

- Sur le côté serveur avec Node.js qui va :
  - Répondre à des routes REST
  - Faire appel à une base de données gérée par MongoDB pour accéder à des documents gérés dans des collections
  - Opérer des traitements de filtres avant de renvoyer les données sous formes d'objets javascript sérialisés (du JSON)
- Sur le côté client avec Angular 2 qui :
  - Gère un seul module (il y aura plusieurs modules dans le projet à développer)
  - Gère un composant service injecté dans différents composants
  - Gère un composant qui fait office de contrôleur.



### *Parc d'applications*

Chaque dossier pour chacune des applications sera présent dans un dossier commun, ces dossiers posséderont les différents fichiers de configuration afin de paramétrer chaque serveur. Ces derniers diffèrent entre chaque application, ainsi que leur localisation dans chaque dossier.

Un dossier comprenant les scripts pour démarrer une application sera présent avec ceux pour éteindre et mettre à jour une application, sans oublier ceux pour la supervision. Ces scripts pourront s'adapter pour correspondre à une application en question car elles n'ont pas tous la même syntaxe pour le lancement, différents paramètres peuvent entrer en jeu.

Il est donc important d'être modulable du côté de l'application web.

### c) Fiabilité et Vulnérabilité

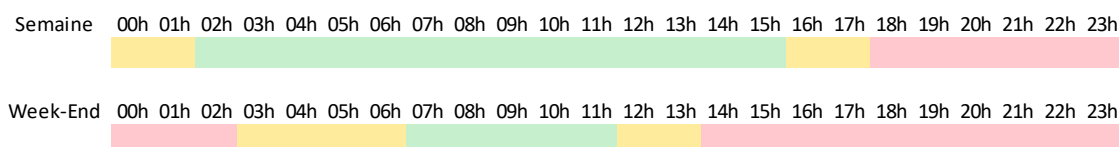
Afin d'augmenter la fiabilité de l'application, des tests unitaires sont mis en place côté back-end et des tests End-to-End du côté du front-end, via l'outil Selenium.

Aussi, la mise en place de SonarQube permettra de tester la qualité du code de l'application web et ainsi en tirer des conclusions dans le but de l'améliorer, cela peut concerner des erreurs liées à l'accessibilité, l'interopérabilité, la performance, la sécurité ou encore les Progressive Web App.

### d) Déploiement

Les outils tels que Jenkins et Docker pourront être utilisés afin de déployer l'application web sur le serveur dédié de manière automatique, afin de réaliser une intégration continue.

### e) Maintenance



Suite à la constitution d'un planning de fréquentation des outils de communication utilisés par l'équipe The Independent Gamers, il est possible de déterminer des zones vertes (fréquentation faible), jaunes (fréquentation modérée) et rouges (fréquentation élevée).

On constate qu'il est préférable de réaliser une maintenance dans le courant de la semaine, ainsi il y'a une plus grande plage horaire ayant une fréquence faible, ce qui permet d'avoir plus de temps pour la résolution d'un problème. Les fréquentations durant un week-end sont similaires à celles qu'on pourrait avoir en semaine durant des vacances ou éventuels congés, il serait donc intéressant de privilégier les mises à jour le matin, dans les alentours de 7h00.

Les mises à jour des applications ne sont pas régulières pour toutes, une date fixe n'est donc pas envisageable. Une vérification de la disponibilité d'une mise à jour peut régulièrement se faire.

### f) Communication

En cas de panne ou un arrêt d'un service, les utilisateurs seront prévenus via un communiqué diffusé via différents moyens de communication (ex : Twitter, Mail, Flux RSS, ...). Si une application doit être mise à jour ou simplement arrêter, les utilisateurs connectés à cette application seront prévenus suivi d'un délai avant l'arrêt du service. Cela s'applique aussi à l'application web.

L'accessibilité à ces services étant restreint, le client s'engage à une communication sur les plans d'action à venir.